Angel Yeremi Ledesma Ayala

Git y GitHub

Pull request:

Un "pull request" (PR) es un mecanismo fundamental en el flujo de trabajo colaborativo de desarrollo de software que utiliza Git y plataformas de alojamiento de código en línea como GitHub. Este proceso permite a los colaboradores proponer cambios en un repositorio de código y solicitar que el propietario del repositorio principal revise y, si es apropiado, integre esos cambios en el proyecto.

A continuación se detalla el proceso de un pull request, junto con los comandos de Git que se utilizan:

1. Creación de una bifurcación (fork):
    - Acción en GitHub: El colaborador comienza por realizar una bifurcación del repositorio principal. Esto se hace en la página del repositorio original en GitHub, haciendo clic en el botón "Fork". Esta acción crea una copia del repositorio en la cuenta del colaborador.
2. Clonación del repositorio:
    - Comando Git: El colaborador clona su repositorio bifurcado en su máquina local utilizando el comando git clone. Por ejemplo:

      git clone https://github.com/cuenta/repositorio.git
3. Creación de una rama:
    - Comandos Git: El colaborador crea una nueva rama local con el comando git branch y se cambia a esa rama utilizando git checkout. Por ejemplo:

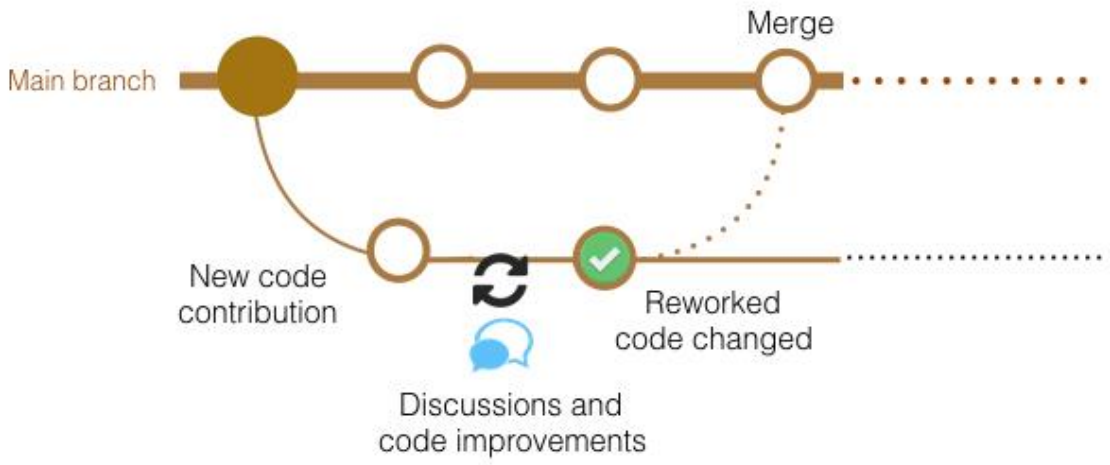
      git branch mi-rama git checkout mi-rama
4. Realización de cambios:
    - Acción local: El colaborador efectúa las modificaciones necesarias en esta rama local. Puede editar, agregar o eliminar archivos según las necesidades del proyecto.
5. Confirmación y envío de cambios:
    - Comandos Git: Los cambios realizados se confirman en la rama local utilizando git commit. Por ejemplo:

      git commit -m "Mensaje explicando el cambio"
6. Creación del pull request:

- Acción en GitHub: Una vez que los cambios están confirmados en la rama, el colaborador regresa a su repositorio bifurcado en GitHub y hace clic en la opción "New Pull Request". Aquí, se proporciona información detallada sobre los cambios y se crea el pull request.

7. Revisión y debate:
    - Acción en GitHub: El propietario del repositorio principal y otros colaboradores revisan el pull request. Pueden dejar comentarios en líneas específicas de código, hacer preguntas o sugerir cambios adicionales. Este proceso fomenta la colaboración y la mejora de la calidad del código.
8. Integración:
    - Acción en GitHub: Si los cambios se consideran adecuados y cumplen con los estándares del proyecto, el propietario del repositorio principal puede "merge" (fusionar) el pull request en el repositorio principal.
9. Cierre del pull request:
    - Acción en GitHub: Después de la fusión exitosa, el pull request se cierra, indicando que las modificaciones propuestas se han incorporado oficialmente al proyecto.

Los pull requests son una parte fundamental del proceso de colaboración en GitHub y otras plataformas similares. Facilitan la revisión de código, la discusión de cambios y proporcionan un registro transparente de la evolución del proyecto.



Simplified Pull Request process

Fork:

La bifurcación es el punto de partida para cualquier colaborador que desee contribuir a un proyecto en GitHub. Este proceso permite a un colaborador crear una copia personal de un repositorio, que será independiente del repositorio original y en la que pueden trabajar libremente sin afectar al proyecto principal.

Pasos para realizar una bifurcación en GitHub:

1. Acceso al repositorio original:
    - El colaborador inicia su contribución visitando el repositorio original en GitHub, que es propiedad del mantenedor del proyecto. Esto se hace navegando a la página del repositorio, que tiene una URL como https://github.com/propiedad-del-mantenedor/repo-original.
2. Bifurcación del repositorio original:
    - En la página del repositorio original, el colaborador hace clic en el botón "Fork" ubicado en la esquina superior derecha de la pantalla. Este botón permite crear una bifurcación del repositorio en la propia cuenta de GitHub del colaborador.
3. Creación de la copia del repositorio:
    - Después de hacer clic en "Fork", GitHub copia todo el repositorio, incluyendo el código, las ramas, la historia de confirmaciones y otros elementos, en la cuenta del colaborador. La URL del nuevo repositorio bifurcado será, por ejemplo, https://github.com/tucuenta/repo-original. Ahora, el colaborador tiene control total sobre esta copia personal.

La bifurcación sirve para varios propósitos fundamentales:

- Desarrollo independiente: El colaborador puede realizar modificaciones en su repositorio bifurcado sin afectar al proyecto original. Esto permite el desarrollo de nuevas características o la corrección de errores sin interferir con el trabajo del mantenedor o de otros colaboradores.
- Aislamiento de cambios: La bifurcación proporciona un entorno aislado para realizar y probar cambios antes de proponerlos al proyecto principal a través de un pull request. Esto ayuda a garantizar que los cambios no causen problemas inesperados.
- Contribución a proyectos de código abierto: Los colaboradores pueden utilizar bifurcaciones para contribuir a proyectos de código abierto en los que no tienen permisos directos. Realizan cambios en su bifurcación y luego crean pull requests para que los mantenedores los revisen e integren en el proyecto principal.

La bifurcación en GitHub es un paso crucial para la colaboración en proyectos de código abierto y el desarrollo colaborativo. Permite a los colaboradores trabajar de manera independiente y contribuir de manera eficiente a proyectos existentes.

Rebase:

El rebase es una operación en Git que se utiliza para reorganizar la historia de confirmaciones de una rama. En lugar de fusionar cambios de una rama en otra (como se hace con git merge), el rebase reorganiza las confirmaciones de la rama actual (rama de trabajo) y las coloca en la parte superior de otra rama, generalmente la rama en la que se desea integrar los cambios. Esto tiene como resultado una historia más lineal y limpia.

Los pasos para realizar un rebase son los siguientes:

1. Actualización de la rama actual:
    - Antes de realizar un rebase, es importante asegurarse de que la rama de trabajo esté actualizada con las últimas confirmaciones del repositorio remoto. Esto se logra con el comando git pull.

      Ejemplo:

      ```
      git pull origin mi-rama
      ```

2. Realización del rebase:
    - Una vez que la rama de trabajo esté actualizada, se puede llevar a cabo el rebase mediante el comando git rebase, especificando la rama a la que se desea reorganizar la rama de trabajo.

      Ejemplo:

      ```
      git rebase rama-objetivo
      ```

Esto implica tomar las confirmaciones de la rama de trabajo, aplicarlas sobre la rama-objetivo y reorganizar la historia de confirmaciones.

3. Resolución de conflictos (si es necesario):
    - Durante el proceso de rebase, pueden surgir conflictos si se han modificado las mismas líneas de código en ambas ramas. Estos conflictos deben ser resueltos manualmente, siguiendo las indicaciones proporcionadas por Git.
4. Continuación del rebase:
    - Tras resolver los conflictos, se puede proseguir con el rebase empleando el comando git rebase --continue.

      Ejemplo:

```
git rebase --continue
```

5. Finalización del rebase:
    - Una vez completado el rebase, se finaliza con git rebase --skip si se desea omitir alguna confirmación en particular o permitir que todas las confirmaciones sean aplicadas.

      Ejemplo:

      ```
      git rebase --skip
      ```

6. Envío de los cambios rebaseados:
    - Al concluir el rebase, es necesario enviar los cambios rebaseados a la rama remota mediante git push.

      Ejemplo:

      ```
      git push origin mi-rama
      ```

El rebase se utiliza comúnmente cuando se trabaja en una rama de características y se busca mantener una historia de confirmaciones más ordenada antes de fusionarla en la rama principal.

Es importante destacar que el rebase es una operación poderosa, pero debe emplearse con precaución, especialmente en ramas compartidas, ya que puede reescribir la historia de confirmaciones, lo que podría ocasionar confusión o conflictos si otros colaboradores trabajan en la misma rama. Por lo tanto, se recomienda la comunicación y coordinación con otros colaboradores al utilizar el rebase.